

CLAIMS

1
2
3 1. A method of operating a filesystem, said filesystem including a live
4 filesystem accessible to users and a zombie filesystem not accessible to users, said method
5 including

6 recording changes to said zombie filesystem in a persistent memory.
7

8 2. A method as in claim 1, including, for a deletion operation on a file
9 in said live filesystem,

10 transferring said file from said live filesystem to said zombie filesystem;

11 breaking links associating disk blocks with said file in a plurality of steps
12 while said file is associated with said zombie filesystem, wherein said recording of
13 changes includes recording said breaking of links in a plurality of steps; and

14 altering said live filesystem to reflect said deletion operation.
15

16 3. A method as in claim 1, including, for a truncation operation on a
17 file in said live filesystem,

18 transferring at least a portion of said file from said live filesystem to said
19 zombie filesystem;

20 breaking links associating disk blocks with said file in a plurality of steps
21 while a portion of said file is associated with said zombie filesystem, wherein said record-
22 ing of changes includes recording said breaking of links in a plurality of steps; and

1 altering said live filesystem to reflect changes associated with said breaking
2 of links.

3
4 4. A method as in claim 1, including, for an operation apparent to users
5 as substantially atomic, performing said operation in a plurality of steps using said zom-
6 bie filesystem, wherein said recording changes is performed in said persistent memory for
7 each of said plurality of steps.

8
9 5. A method as in claim 1, including, for an operation performed on a
10 file having attached data elements, performing said operation using said zombie
11 filesystem.

12
13 6. A method as in claim 1, including, for an operation performed using
14 said zombie filesystem, altering a size of said zombie filesystem during performance of said
15 operation.

16
17 7. A method as in claim 1, including, for an operation performed using
18 said zombie filesystem, checkpointing said filesystem during performance of said opera-
19 tion.

1 8. A method as in claim 1, including recording changes to said live
2 filesystem in said persistent memory, wherein records of changes to said live filesystem and
3 of changes to said zombie filesystem are substantially interspersed.

4
5 9. A method as in claim 1, including replaying a set of said changes in
6 response to said record.

7
8 10. A method as in claim 1, including replaying a set of said changes to
9 said live filesystem and to said zombie filesystem, wherein replay of changes includes sub-
10 stantial interspersed performance of changes to said live filesystem and to said zombie
11 filesystem.

12
13 11. A method as in claim 1, including replaying a set of said changes in
14 said record in response to a crash recovery by said filesystem.

15
16 12. A method as in claim 1, wherein said persistent memory includes a
17 log of substantially all changes, within a selected time duration, to either said live
18 filesystem or said zombie filesystem.

19
20 13. A method as in claim 1, wherein said persistent memory includes a
21 log of substantially all changes, within a selected time duration, to said zombie filesystem.

1 14. A method as in claim 1, wherein said recorded changes include a set
2 of substantially atomic operations to said zombie filesystem.

3
4 15. A method of operating a filesystem, said filesystem including a live
5 filesystem accessible to users and a zombie filesystem not accessible to users, said method
6 including
7 dynamically growing said zombie filesystem.

8
9 16. A method as in claim 15, including, for a deletion or truncation op-
10 eration on a file in said live filesystem,
11 allocating storage within said zombie filesystem for metadata associated with
12 said file;
13 performing said dynamic growth in response to failure of said allocation of
14 storage;
15 re-performing said allocation of storage after said dynamic growth; and
16 transferring said file from said live filesystem to said zombie filesystem.

17
18 17. A method as in claim 15, wherein said dynamic growth occurs, for
19 an operation performed using said zombie filesystem, during performance of said opera-
20 tion.

1 18. A method of operating a filesystem, said filesystem including a live
2 filesystem accessible to users and a zombie filesystem not accessible to users, said method
3 including

4 transfer of a file to said zombie filesystem before breakage of links to blocks
5 in said file, in response to an operation on said file, said operation using said zombie
6 filesystem.

7
8 19. A method as in claim 18, wherein, for a deletion operation on a file
9 in said live filesystem,

10 said transfer includes

11 creating a link associating said file with said zombie filesystem; and

12 breaking a link associating said file with said live filesystem;

13 and said deletion operation includes

14 breaking links associating disk blocks with said file in a plurality of steps
15 while said file is associated with said zombie filesystem, wherein said recording of
16 changes includes recording said breaking of links in a plurality of steps; and

17 altering said live filesystem to reflect said deletion operation.

18
19 20. A method as in claim 18, wherein, for a truncation operation on a
20 file in said live filesystem,

21 said transfer includes

1 creating a link associating at least a portion of said file with said zombie
2 filesystem; and
3 breaking a link associating said portion with said file in said live filesystem;
4 and said truncation operation includes
5 breaking links associating disk blocks with said file in a plurality of steps
6 while a portion of said file is associated with said zombie filesystem, wherein said record-
7 ing of changes includes recording said breaking of links in a plurality of steps; and
8 altering said live filesystem to reflect changes associated with said breaking
9 of links.

11 21. A method of operating a filesystem, said filesystem including a live
12 filesystem accessible to users and a zombie filesystem not accessible to users, said method
13 including

14 transfer of a file to said zombie filesystem before performing any substantial
15 portion of an operation on said file, said operation using said zombie filesystem.

17 22. A method as in claim 21, wherein, for a deletion operation on a file
18 in said live filesystem,

19 said transfer includes

20 creating a link associating said file with said zombie filesystem; and

21 breaking a link associating said file with said live filesystem;

22 and said deletion operation includes

1 breaking links associating disk blocks with said file in a plurality of steps
2 only while said file is associated with said zombie filesystem, wherein said recording of
3 changes includes recording said breaking of links in a plurality of steps; and
4 altering said live filesystem to reflect said deletion operation.
5

6 23. A method as in claim 21, wherein, for a truncation operation on a
7 file in said live filesystem,

8 said transfer includes
9 creating a link associating at least a portion of said file with said zombie
10 filesystem; and

11 breaking a link associating said portion with said file in said live filesystem;
12 and said truncation operation includes

13 breaking links associating disk blocks with said file in a plurality of steps
14 only while a portion of said file is associated with said zombie filesystem, wherein said re-
15 cording of changes includes recording said breaking of links in a plurality of steps; and

16 altering said live filesystem to reflect changes associated with said breaking
17 of links.
18

19 24. A method of operating a filesystem, said filesystem including a live
20 filesystem accessible to users and a zombie filesystem not accessible to users, said method
21 including

22 replay of an operation on a file, said operation using said zombie filesystem.

1
2 25. A method as in claim 24, wherein said replay is responsive to a set
3 of recorded changes in a persistent memory;
4 and including, for a deletion operation on a file in said live filesystem,
5 transferring said file from said live filesystem to said zombie filesystem, and
6 recording said transfer in said persistent memory;
7 breaking links associating disk blocks with said file in a plurality of steps
8 while said file is associated with said zombie filesystem, and recording said breaking of
9 links in said persistent memory in a plurality of steps; and
10 altering said live filesystem to reflect said deletion operation, and recording
11 said alteration in said persistent memory.

12
13 26. A method as in claim 24, wherein said replay is responsive to a set
14 of recorded changes in a persistent memory;
15 and including, for a truncation operation on a file in said live filesystem,
16 transferring at least a portion of said file from said live filesystem to said
17 zombie filesystem, and recording said transfer in said persistent memory;
18 breaking links associating disk blocks with said file in a plurality of steps
19 while a portion of said file is associated with said zombie filesystem, and recording said
20 breaking of links in said persistent memory in a plurality of steps; and
21 altering said live filesystem to reflect changes associated with said breaking
22 of links, and recording said alteration in said persistent memory.

1
2 27. A method of operating a filesystem, said filesystem including a live
3 filesystem accessible to users and a zombie filesystem not accessible to users, said method
4 including
5 replay of a set of filesystem operations, said operations including at least
6 some operations using said live filesystem and at least some operations using said zombie
7 filesystem.

8
9 28. A method as in claim 27, wherein said replay is responsive to a set
10 of recorded changes in a persistent memory;

11 and including, for a deletion operation on a file in said live filesystem,
12 transferring said file from said live filesystem to said zombie filesystem, and
13 recording said transfer in said persistent memory;

14 breaking links associating disk blocks with said file in a plurality of steps
15 while said file is associated with said zombie filesystem, and recording said breaking of
16 links in said persistent memory in a plurality of steps; and

17 altering said live filesystem to reflect said deletion operation, and recording
18 said alteration in said persistent memory.

19
20 29. A method as in claim 27, wherein said replay is responsive to a set
21 of recorded changes in a persistent memory;

22 and including, for a truncation operation on a file in said live filesystem,

1 transferring at least a portion of said file from said live filesystem to said
2 zombie filesystem, and recording said transfer in said persistent memory;
3 breaking links associating disk blocks with said file in a plurality of steps
4 while a portion of said file is associated with said zombie filesystem, and recording said
5 breaking of links in said persistent memory in a plurality of steps; and
6 altering said live filesystem to reflect changes associated with said breaking
7 of links, and recording said alteration in said persistent memory.

FILED
U.S. DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK
2020 APR 14 PM 4:05
CLERK OF COURT